

**IN THE CLAIMS:**

1. (Currently Amended): A method on a server in a distributed data processing system for maintaining a logical composite repository of Object Identifier (OID) tree structures, the method comprising the steps of:

receiving, in an OID abstraction layer, an OID tree structure from a repository; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

registering the OID tree structure with a registry associated with the OID abstraction layer; and

adding the OID tree structure to a repository associated with the OID abstraction layer.

2. (Original): The method of claim 1, wherein the registry associated with the OID abstraction layer provides information identifying an anchor point in the OID subtree structure to be maintained by the repository.

3. (Original): The method of claim 2, wherein if the anchor point of the OID subtree structure is already registered with the OID abstraction layer, the registry is overwritten.

4. (Original): The method of claim 2, wherein if a query is received for an object that has an Object Identifier that is below a registered anchor point in an OID tree structure, the OID abstraction layer identifies a repository that maintains object information for the requested object based on the registered anchor point.

5. (Original): The method of claim 1, wherein the repository is configured such that the repository recognizes requests from an application program interface (API) associated with the OID abstraction layer and sends reply messages to the API containing information retrieved from the repository.

6. (Currently Amended): The method of claim 5, wherein the OID abstraction layer receives the information retrieved from the repository through the API and encapsulates the information in a reply message to a target protocol interface, wherein the reply message is formatted for an appropriate protocol for the target protocol interface, and wherein the appropriate protocol is one of the two or more different protocols.

7. (Currently Amended): The method of claim 1, wherein the OID abstraction layer receives a request for object data from a requesting protocol interface, interprets the request according to a protocol of the requesting protocol interface, wherein the protocol of the requesting protocol interface is one of the two or more different protocols, converts the request into an application program interface (API) request that which is forwarded to the repository, and receives an API reply from the repository having the object data.

8. (Currently Amended): The method of claim 7, wherein the OID abstraction layer reformats the object data in a reply message according to the protocol of the requesting protocol interface and sends the reply message to the requesting protocol interface.

9. (Currently Amended): A method on a server in a distributed data processing system for retrieving object data from a repository, comprising:

receiving a first query for the object data from a requester in the distributed data processing system, wherein the first query is in a protocol recognized by an OID abstraction layer; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

interpreting the first query according to the protocol recognized by the OID abstraction layer, wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols;

locating a repository that contains the object data requested in the first query based on a registry associated with the OID abstraction layer; and

retrieving the object data from the repository using an OID abstraction layer application program interface (API).

10. (Original): The method of claim 9, wherein the first query is mapped into a second query, wherein the second query is consistent with an application program interface (API) associated with the OID abstraction layer.

11. (Original): The method of claim 10, wherein if the first query cannot be mapped into a second query due to a limitation of the repository that contains the object associated with the first query, then the first query cannot be satisfied.

12. (Original): The method of claim 10, wherein the second query is sent to the repository that contains the object associated with the first query.

13. (Original): The method of claim 12, wherein a first reply is received at the API associated with the OID abstraction layer from the repository that contains the object associated with the first query.

14. (Currently Amended): The method of claim 13, wherein the first reply is transformed into a second reply, wherein the second reply is consistent with the protocol for the first query recognized by the OID abstraction layer, and wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols.

15. (Original): The method of claim 14, wherein the second reply is sent to the requester in the distributed data processing system.

16. (Currently Amended): The method of claim 9, wherein each repository in [[the]] a plurality of repositories contains information representing an Object Identifier (OID) subtree structure, and wherein the plurality of repositories are formatted to support the two or more different protocols.

17. (Original): The method of claim 9, wherein Simple Network Management Protocol (SNMP) is a protocol recognized by the OID abstraction layer.

18. (Original): The method of claim 9, wherein Lightweight Directory Access Protocol (LDAP) is a protocol recognized by the OID abstraction layer.

19. (Original): The method of claim 9, wherein Common Information Model used in conjunction with eXtendable Markup Language (CIM/XML) is a protocol recognized by the OID abstraction layer.

20. (Currently Amended): An apparatus on a server in a distributed data processing system for maintaining a logical composite repository of Object Identifier (OID) tree structures, the apparatus comprising:

an OID abstraction layer that receives an OID tree structure from a repository; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

a registry, associated with the OID abstraction layer, that registers the OID tree structure; and

an adding means for adding the OID tree structure to a repository associated with the OID abstraction layer.

21. (Original): The apparatus of claim 20, wherein the registry provides information identifying an anchor point in the OID tree structure to be maintained by the repository.

22. (Original): The apparatus of claim 21, wherein if the anchor point of the OID tree structure is already registered in the registry, then the registry overwrites the previous entry.

23. (Original): The apparatus of claim 21, wherein, if the OID abstraction layer receives a query for an object that has an Object Identifier that is below a registered anchor point in an OID tree structure, the registry in the OID abstraction layer identifies a repository that maintains object information for the requested object based on the registered anchor point.

24. (Original): The apparatus of claim 20, wherein the repository is configured such that the repository recognizes requests received from an application program interface (API) associated with the OID abstraction layer and sends reply messages to the API containing information retrieved from the repository.

25. (Currently Amended): The apparatus of claim 24, wherein the OID abstraction layer receives the information retrieved from the repositories through the API and encapsulates the information in a reply message to a target protocol interface, wherein the reply message is formatted for an appropriate protocol for the target protocol interface, and wherein the appropriate protocol is one of the two or more different protocols.

26. (Currently Amended): The apparatus of claim 20, wherein the OID abstraction layer receives a request for object data from a requesting protocol interface, interprets the request according to a protocol of the requesting protocol interface, wherein the protocol of the requesting protocol interface is one of the two or more different protocols, converts the request into an application program interface (API) request that which is forwarded to the repository, and receives an API reply from the repository having the object data.

27. (Currently Amended): The apparatus of claim 26, wherein the OID abstraction layer encapsulates the object data in a reply message according to the protocol of the requesting protocol interface and sends the reply message to the requesting protocol interface.

28. (Currently Amended): An apparatus on a server in a distributed data processing system for retrieving object data from a repository, comprising:

a receiving means for receiving a first query for the object data from a requester in the distributed data processing system, wherin the first query is in a protocol recognized by an OID abstraction layer; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

a interpreting means for interpreting the first query according to the protocol recognized by the OID abstraction layer, wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols;

a locating means for locating a repository that contains the object data requested in the first query based on a registry associated with the OID abstraction layer; and

a retrieving means for retricving the objcct data from the repository using an OID abstraction layer application program interface (API).

29. (Original): The apparatus of claim 28, further comprising a mapping means for mapping the first query into a second query, wherein the second query is consistent with an application program interface (API) associated with the OID abstraction layer.

30. (Original): The apparatus of claim 29, wherein if the mapping means cannot map the first query into a second query due to a limitation of the repository that contains the object associated with the first query, then the first query cannot be satisfied.

31. (Original): The apparatus of claim 29, further comprising a first sending means, in the OID abstraction layer, that sends the second query to a repository that contains the object associated with the first query.

32. (Original): The apparatus of claim 31, wherin the retrieving means receives a first reply at the API from the repository that contains the object associated with the first query.

33. (Currently Amended): The apparatus of claim 32, further comprising a transforming means, in the OID abstraction layer, that transforms the first reply into a second reply, wherein the second reply is consistent with the protocol for the first query recognized by the OID abstraction layer, and wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols.

34. (Original): The apparatus of claim 33, further comprising a second sending means, in the OID abstraction layer, that sends the second reply to the requester in the distributed data processing system.

35. (Currently Amended): The apparatus of claim 28, wherein each repository in a plurality of repositories contains Object Identifier (OID) tree structures, and wherein the plurality of repositories are formatted to support the two or more different protocols.

36. (Original): The apparatus of claim 28, wherein the receiving means recognizes a Simple Network Management Protocol (SNMP) query.

37. (Original): The apparatus of claim 28, wherein the receiving means recognizes a Lightweight Directory Access Protocol (LDAP) query.

38. (Original): The apparatus of claim 28, wherein the receiving means recognizes a Common Information Model used in conjunction with eXtendable Markup Language (CIM/XML) query.

39. (Currently Amended): A computer program product in a computer readable medium for maintaining a repository of Object Identifier (OID) tree structures, comprising:

instructions for receiving, in an OID abstraction layer, an OID tree structure from a repository; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

instructions for registering the OID tree structure with a registry associated with the OID abstraction layer; and

instructions for adding the OID tree structure to a repository associated with the OID abstraction layer.

40. (Original): The computer program product of claim 39, further comprising instructions for maintaining the registry associated with the OID abstraction layer and providing information identifying an anchor point in the OID tree structure to be maintained by the repository.

41. (Original): The computer program product of claim 40, wherein if the anchor point of the OID tree structure is already registered with the OID abstraction layer, the instructions for registering overwrites the previous entry.

42. (Original): The computer program product of claim 40, further comprising instructions for identifying a repository that maintains object information for the requested object based on the registered anchor point if a query is received for an object that has an Object Identifier that is below a registered anchor point in an OID tree structure.

43. (Original): The computer program product of claim 39, further comprising instructions for configuring the repository to recognize requests from an application program interface (API) associated with the OID abstraction layer and to send reply messages to the API containing information retrieved from the repository.

44. (Currently Amended): The computer program product of claim 43, further comprising instructions for receiving the information retrieved from the repository, through the API, and encapsulating the information in a reply message to a target protocol interface, wherin the reply message is formatted for an appropriate protocol for the target protocol interface, and wherein the appropriate protocol is one of the two or more different protocols.

45. (Currently Amended): The computer program product of claim 39, further comprising:

instructions for receiving a request for object data from a requesting protocol interface;

instructions for interpreting the request according to a protocol of the requesting protocol interface, wherein the protocol of the requesting protocol interface is one of the two or more different protocols,

instructions for converting the request into an application program interface (API) request which is forwarded to the subtree repository; and

instructions for receiving an API reply from the subtree repository having the object data.

46. (Currently Amended): The computer program product of claim 45, further comprising instructions for encapsulating the object data in a reply message according to the protocol of the requesting protocol interface and sending the reply message to the requesting protocol interface.

47. (Currently Amended): A computer program product in a computer readable medium for retrieving object data from a repository, comprising:

instructions for receiving a first query for the object data from a requester in the distributed data processing system, wherein the first query is in a protocol recognized by an OID abstraction layer; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

instructions for interpreting the first query according to the protocol recognized by the OID abstraction layer, wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols;

instructions for locating a repository that contains the object data requested in the first query based on a registry associated with the OID abstraction layer; and

instructions for retrieving the object data from the repository using an OID abstraction layer application program interface (API).

48. (Original): The computer program product of claim 47, wherein the instructions for receiving the first query map the first query into a second query, wherein the second query is consistent with an application program interface (API) associated with the OID abstraction layer.

49. (Original): The computer program product of claim 48, wherein if the instructions for receiving the first query map cannot map the first query into a second query due to a limitation of the repository that contains the object associated with the first query, then the first query cannot be satisfied.

50. (Original): The computer program product of claim 48, further comprising instructions for sending the second query to the repository that contains the object associated with the first query.

51. (Original): The computer program product of claim 50, further comprising instructions for receiving a first reply at the API associated with the OID abstraction layer from the repository that contains the object associated with the first query.

52. (Currently Amended): The computer program product of claim 51, further comprising instructions for transforming the first reply into a second reply, wherein the second reply is consistent with the protocol for the first query recognized by the OID abstraction layer, and wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols.

53. (Original): The computer program product of claim 52, further comprising instructions for sending the second reply to the requester in the distributed data processing system.

54. (Currently Amended): The computer program product of claim 47, wherein [[the]] each repository in a plurality repositories contains Object Identifier (OID) tree structures, and wherein the plurality of repositories are formatted to support the two or more different protocols.

55. (Original): The computer program product of claim 47, wherein instructions for receiving a first query recognize a Simple Network Management Protocol (SNMP) query.

56. (Original): The computer program product of claim 47, wherein instructions for receiving a first query recognize a Lightweight Directory Access Protocol (LDAP) query.

57. (Original): The computer program product of claim 47, wherein instructions for receiving a first query recognize a Common Information Model used in conjunction with eXtendable Markup Language (CIM/XML) query.